2nd Annual GSFC-JPL Quality Mission Software Workshop

Goddard Space Flight Center
Jet Propulsion Laboratory

Session 3: Metrics

San Diego, California
May 16-18, 2000

AGENDA
Session 3: Metrics

Day 2
Wednesday – May 17, 2000

Metrics

8:30 pm	Who Measures and Why	(Presentation & Discussion)	M. Stark
9:20 pm	Strategies for Metrics Infusion	(Presentation & Discussion)	M. Stark
10:00 pm	Break		
10:15 pm	Choosing From Available Metrics	(Presentation & Discussion)	J. Kelly
11:30 pm	Independent Verification and Validation (IV&V) Facility		T. Hammer
Noon	Lunch		

QMS Metrics Workshop

John Kelly & Mike Stark

May 17, 2000

Agenda for Metrics Workshop

- Topic 1: Who measures & why?
- Workshop Part 1
- Topic 2: Strategies for metrics infusion
- Workshop Part 2
- Topic 3: Choosing from available metrics
- Workshop Part 3

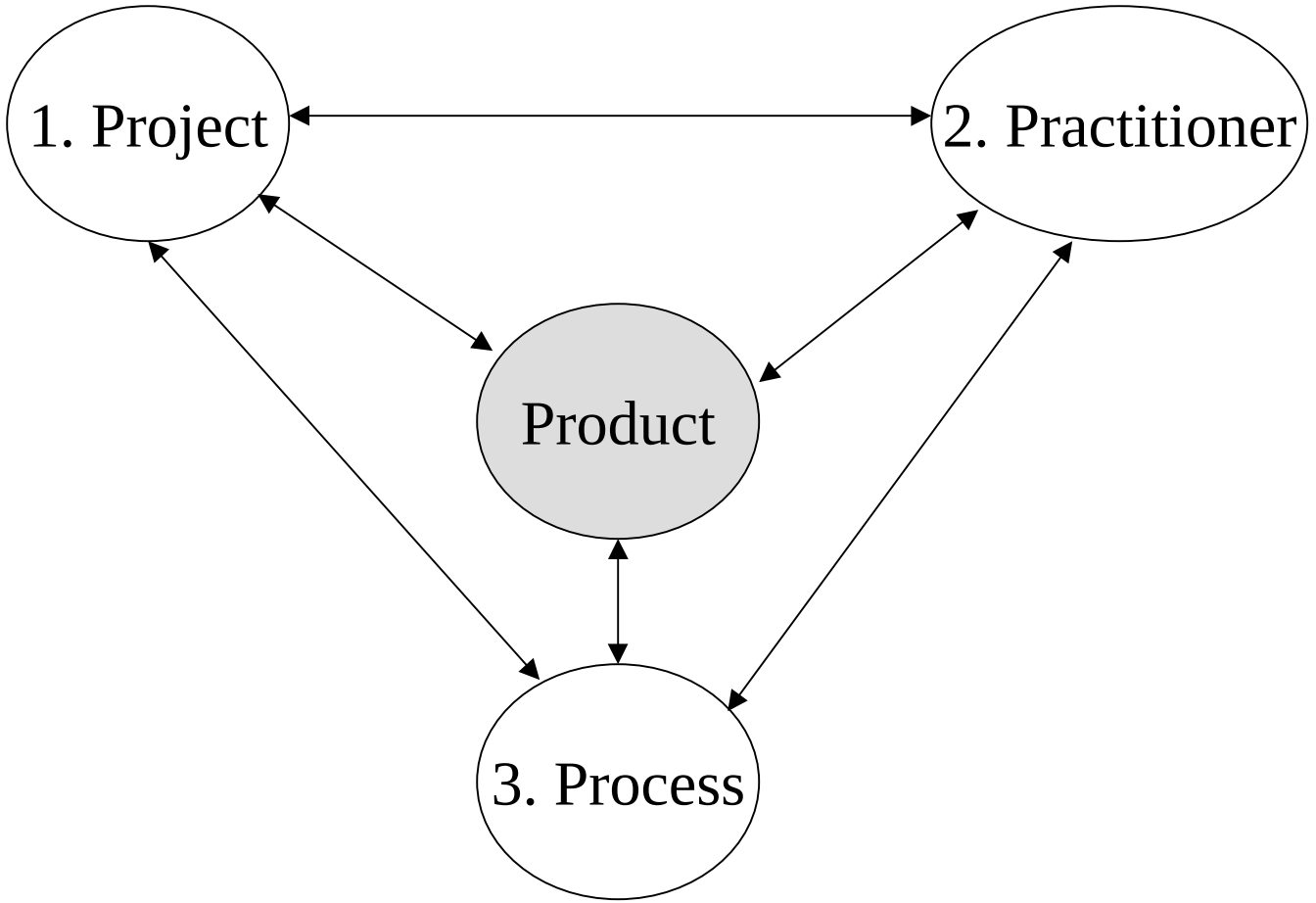
2nd Annual GSFC-JPL Quality Mission Software Workshop

Metrics
Who Measures and Why?

Mike Stark & John Kelly

May 17, 2000

Who Measures? QMS Perspectives



Why Measure? NASA Core Metrics

Draft (1 of 2)

- Building a baseline
  - Models such as effort distribution, error classifications
  - Used as basis of comparison
- Project management
  - Compare actual and expected behavior
  - Identify timely corrective actions

Why Measure? NASA Core Metrics

Draft (2 of 2)

- Quality assurance
  - Assess products for acceptable quality
  - Metrics include requirements quality, code complexity, defect rates,…
  - Values of metrics are used to determine risk
- Product improvement
  - Use metrics to determine where improvement is needed
  - Assess impact of process change on the product

Why Measure? Variations on a Theme

- Other sources
  - *ISC Product Development Handbook*: doesn't discuss baseline models, briefly mentions project management, QA, and improvement
  - *NASA Measurement Guidebook*: doesn't include QA, discusses management, improvement, and baseline models ("understanding") in more detail

"Most failed measurement programs suffer from inadequate or unclear use of data, not from an inadequate or unclear data collection process"
— NASA Measurement Guidebook, p. 6

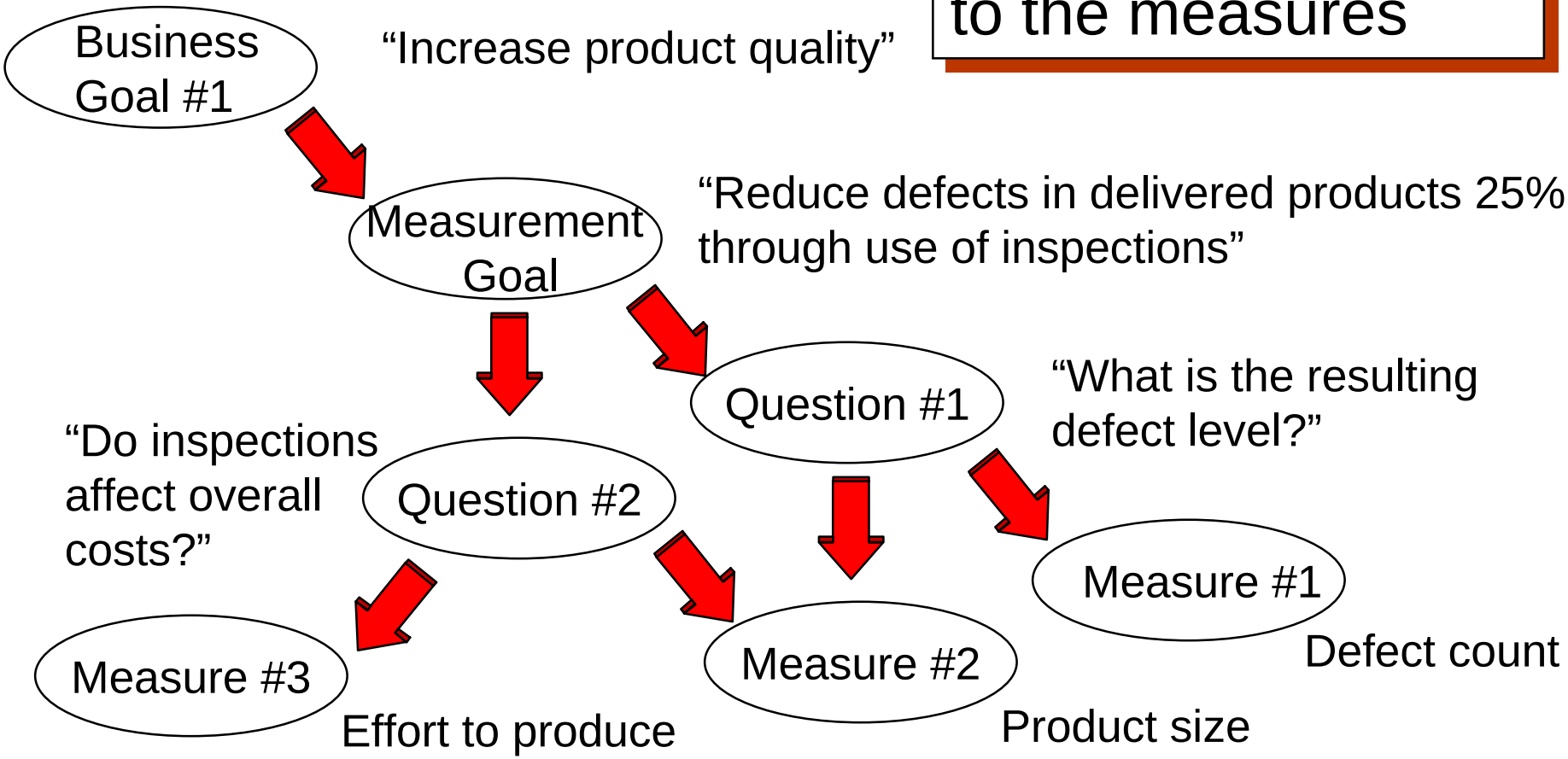
Goal/Question/Metric Paradigm

- ***GQM is an approach that assures data collected are consistent with goals***

Step	Activity	Example
Business Goal	Define desired improvements	“Increase product reliability”
Measurement Goal	Select a change to evaluate & define expectations	“Decrease errors by 25% via inspections w/ no cost increase”
Questions	Define criteria for success or failure	“What are current and new error rates?” “Do inspections add cost?”
Metrics	Define data needed to answer questions	Error counts, system size, staff hours

Goal, Question, Metric Example

GQM links the goals to the measures



Why GQM??

- Align measurement with organization's needs (business goals)
  - Organization can be project,division, research group, etc.
- Maximize information derived from metrics
- Minimize data collection overhead

"Most failed measurement programs suffer from inadequate or unclear use of data, not from an inadequate or unclear data collection process"
— NASA Measurement Guidebook, p. 6

Workshop Part 1: Setting Business Goals

- Step 1: For each of project,practitioner, and process, identify roles and perspectives within the group.
- Step 2: For each role identified, what are their business goals? *These do not have to be stated quantitatively yet*!
- Notes:
  - we may need to look at a subset of perspectives
  - Think about the product as we do this

Example: Marti Szczur

- Role: practitioner/line management
- Business goals
  - Bug free software
  - Accurate cost and schedule estimation
  - Cost and schedule reduction

Step 1: Roles & Perspectives

- For project
- For practitioner
- For process

Step 2: Business Goals

- For project
- For practitioner
- For process

2nd Annual GSFC-JPL Quality Mission Software Workshop

Strategies for Metrics Infusion
or
Why do people hate metrics and what can be done about it?

Mike Stark & John Kelly

May 17, 2000

Why do developers hate metrics?

- Data are collected, no feedback is provided
- Data not intended for project use (e.g. used for improvement, not project management)
- “Any data you provide can be used against you in a court of law (or a performance review)”
- Data collection perceived as redundant
- Lack of knowledge/training on interpretation of data

What can be done about it?

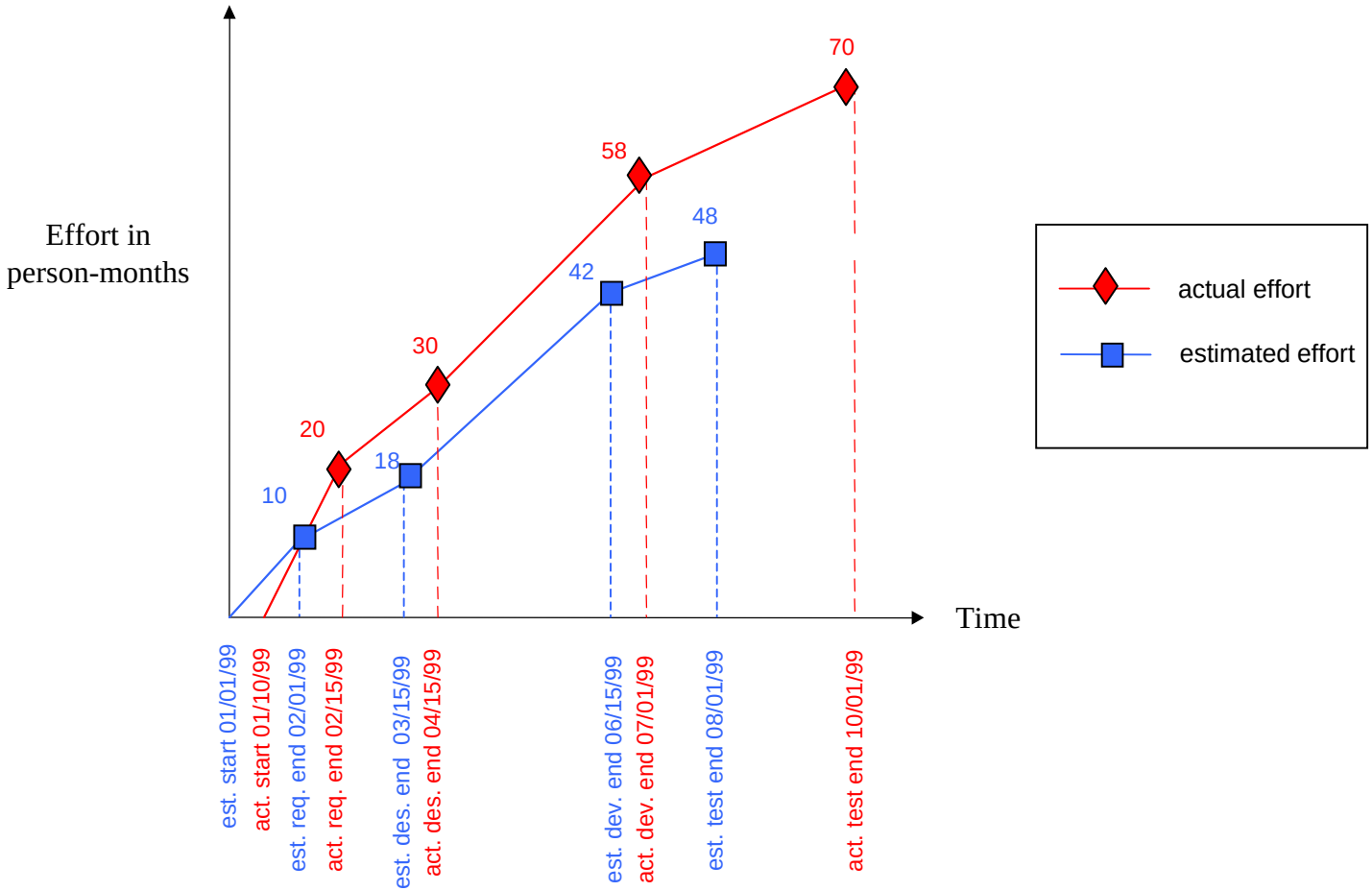
- Improved tools for development teams
- Education on need for and use of data
- Management enforcement

Management enforcement may be necessary,
But it is NOT sufficient

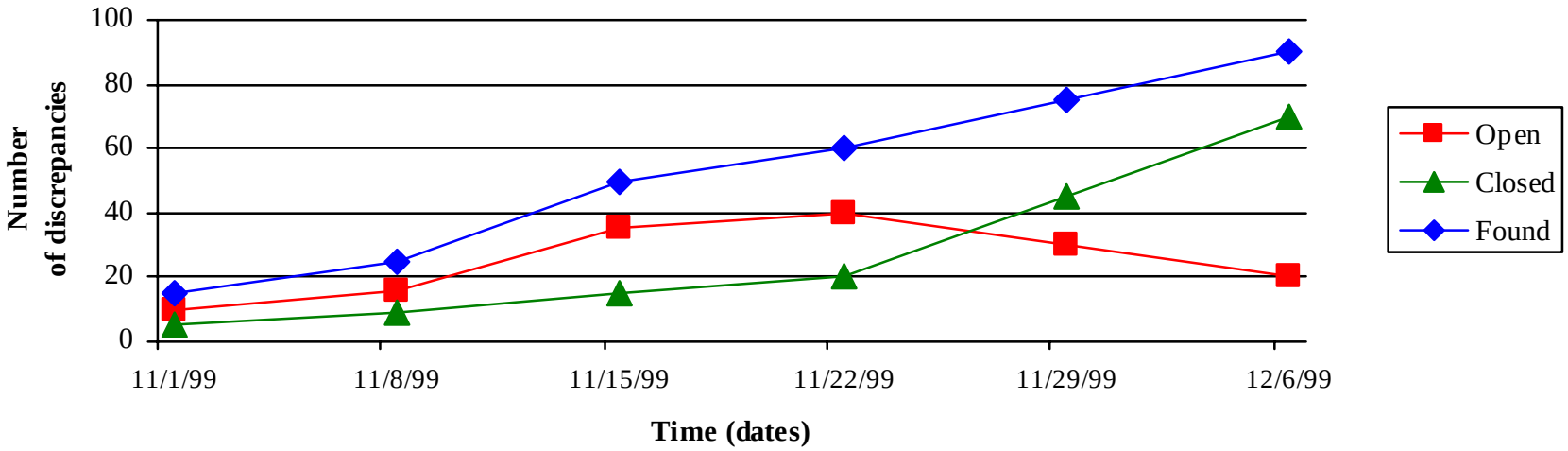
SEL Data Collection and Reporting

- DC & RS: A tool that will provide:
  - Reports on demand via the Web
    - For development team, not line management
  - Contains all metrics required for ISO9000 and generates appropriate reports
  - Reports designed for easy interpretation
  - Data collection integrated with corresponding development/management activities
    - Requirements being generated via use case analysis

Effort and Schedule Tracking



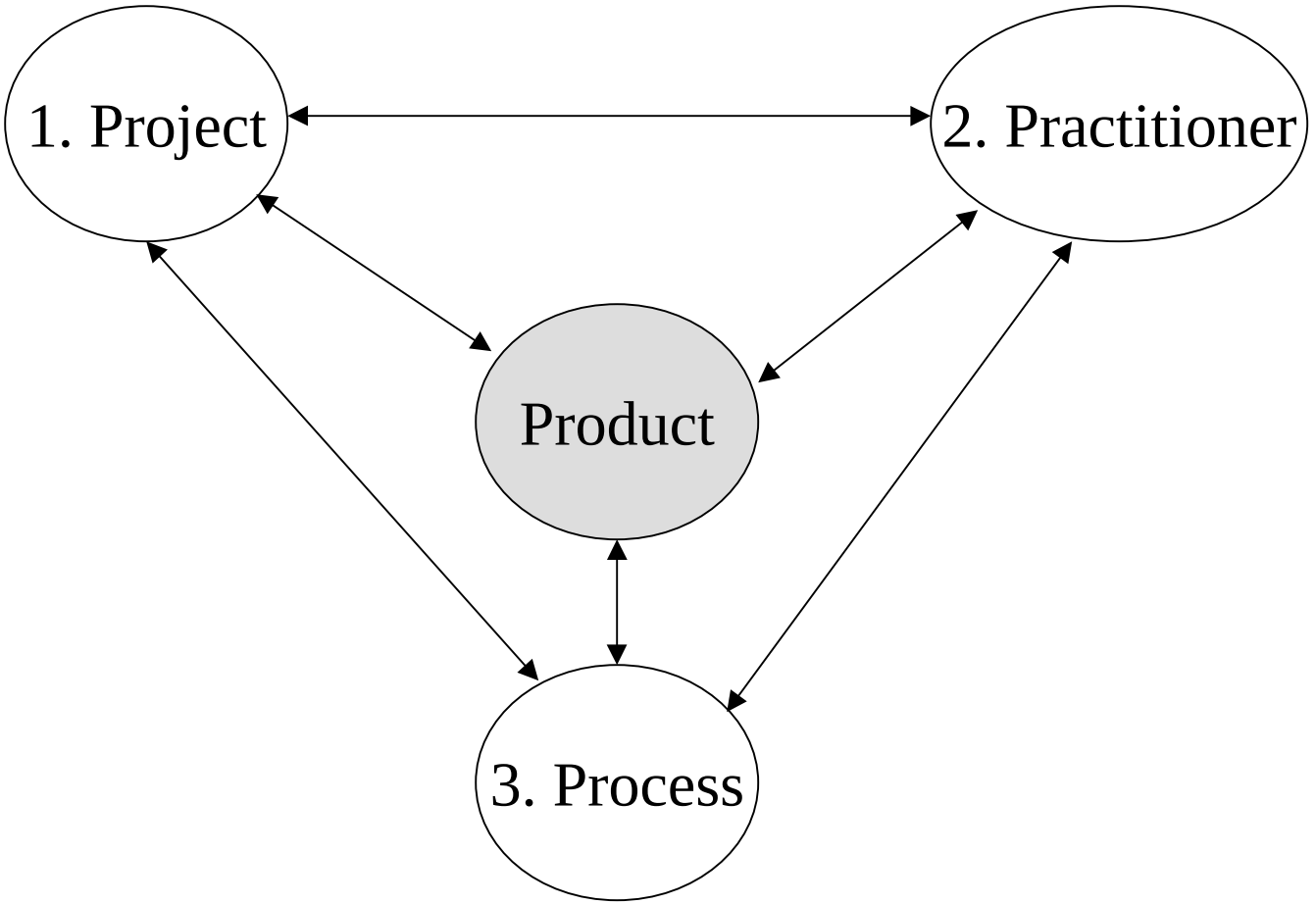
Cumulative Open Discrepancies vs. Closed Discrepancies



Example Scenario

- User is team lead
- Examines effort/schedule chart, sees that testing is slipping
- Looks at discrepancy chart to see if they are accumulating too much
  - Ultimately, you would be able to compare error rates to historic data (previous similar projects)
  - Would see accumulating discrepancies earlier in project and avoid testing delays

QMS Perspectives



Workshop Part 2: Infusion strategies

- Step 3: For each of the roles and perspectives identified in Part 1, identify obstacles to using metrics
- Step 4: For each obstacle, identify a strategy to overcome it
- Notes:
  - Talk shows some of possibilities for practitioner
  - Many possible roles for each “3 P” perspective
  - Think about how 3 P’s interact

Step 3: Identify obstacles to metrics usage

- For project
- For practitioner
- For process

Step 4: Overcoming the obstacles

- For project
- For practitioner
- For process

2nd Annual GSFC-JPL Quality Mission Software Workshop

Metrics:
Choosing From Available Metrics

John Kelly, Mike Stark & Carmen Mikulski

May 16 -18, 2000

Objectives

- Summarize the three levels of software metrics sets:
  - Standards, Policies & Guidelines (Highest Level)
  - General Metrics Sets
  - Focused or Specialized Metrics Sets (Detailed)
- Discuss “General Metrics Sets” in the context of the GQM approach and the Draft NASA Core Metrics Set
- Workshop 3a: Prioritize a subset of the NASA Core Metrics that match the goals from Workshop 1
- Discuss “Focused or Specialized Metrics sets” in the context of Software Inspections
- Workshop 3b: Outline a strategy for implementing software metrics programs with the assistance of Goddard & JPL “championing groups” (i.e. SEL, SATC, & MSP)

Outline

I. Overview of relevant Metrics Standards, Policies & Guidelines (10 mins)

II. Metrics Sets (40 mins)

- NASA Core Metrics Set draft, PSM, NASA Measurement Handbook, ...
- Discussion & Workshop Exercise

III. Focused or Specialized Metrics (20 mins)

- Software Inspections (Lessons Learned)
- Reliability
- ODC

IV. Outlining a Metrics Strategy (Discussion & Workshop) (30 mins)

I. NASA Software Policy NPD 2820.1 (Section 7)

"a. The following shall be evaluated for compliance with this Directive:

(1) Evidence of project conformance to this policy as reported to the governing PMC.

(2) Agency trends on the following:

(a) Software cost and schedule baseline deviations; and

(b) Degree to which delivered software satisfies its requirements, including safety, quality, and reliability measures.

(3) Results of the following:

(a) Assessments and audits of conformance to ISO 9001 and the CMM in NASA software creation and acquisition organizations;

(b) Other surveys relating to the implementation of this Directive;

(c) Improvements in software acquisition and creation on software projects, resulting from the use of the CMM;

(d) Improvements in management of software creation and acquisition, resulting from case studies and shared experiences.

b. Specific responsibilities for collecting, analyzing, and reporting metrics are contained in NPG 2820*."

*Note: An NPD 2820 Metrics Report, Jan 2000 exists and a proposal to create a NPG 2820 has been submitted to the recent AE Call for Proposals, but there currently is not a NPG

I. NASA Senior Management Council Presentation by Lee Holcomb (NASA CIO) April, 2000

Metrics for Software Development Activities (Quarterly)

Cost	Plan vs. Actual
Schedule	Plan vs. Actual
Workforce	Plan vs. Actual
Requirements	Percent Change Since Baseline
Development	Plan vs. Actual *
Testing	Number of Open/Closed DR's

*Note: Actual work units completed (i.e. units successfully designed, coded, integrated and tested)

I. THE JPL SOFTWARE DEVELOPMENT PROCESS DESCRIPTION, JPL D-15378, Rev. D

- ***"Metrics, and the associated procedures for collecting, storing, and analyzing them, shall be identified in a development plan, and shall be tailored to project needs.*** *[6.4.1][ISO 4.16]"*
- Recommended metrics are:
  - Deviations from the staffing plan, development milestones, and budget (planned vs. actual); deviations from projected earned value should be considered.
  - Comparison of planned and actual output, typically done in terms of thousands of lines of source code (KSLOC), function points, or module count; projection of output required to complete the product. An automated tool should be used for KSLOC counts.

I. THE JPL SOFTWARE DEVELOPMENT PROCESS DESCRIPTION, JPL D-15378, Rev. D

- Analysis of anomalies or defects during development; plots of anomalies identified and corrected are customarily used during development to assess build stability, and readiness for acceptance testing and delivery. Analysis of defect data may also be used to find and correct problems in the development process.
- Number of changes in required product capabilities after the requirements were baselined.
- Elapsed time between the identification and resolution of a problem, by development phase.

II. Metrics Sets

These are metrics sets which address general concerns regarding the development and operation of software

Examples:

- Draft NASA Core Metrics (Jan, 2000)
- Practical Software Measurement (PSM) - DoD
- ARMY STEP Metrics
- SEI CMM Measures

II. Practical Software Measurement (PSM)

- Tailor a measurement set for each task based on the particular task's issues
- Provide a set of 7 common software issues:
  - Schedule and Progress
  - Resources and Cost
  - Product Size and Stability
  - Product Quality
  - Process Performance
  - Technology Effectiveness
  - Customer Satisfaction
- Provide various measures examples for each of the Issue Categories

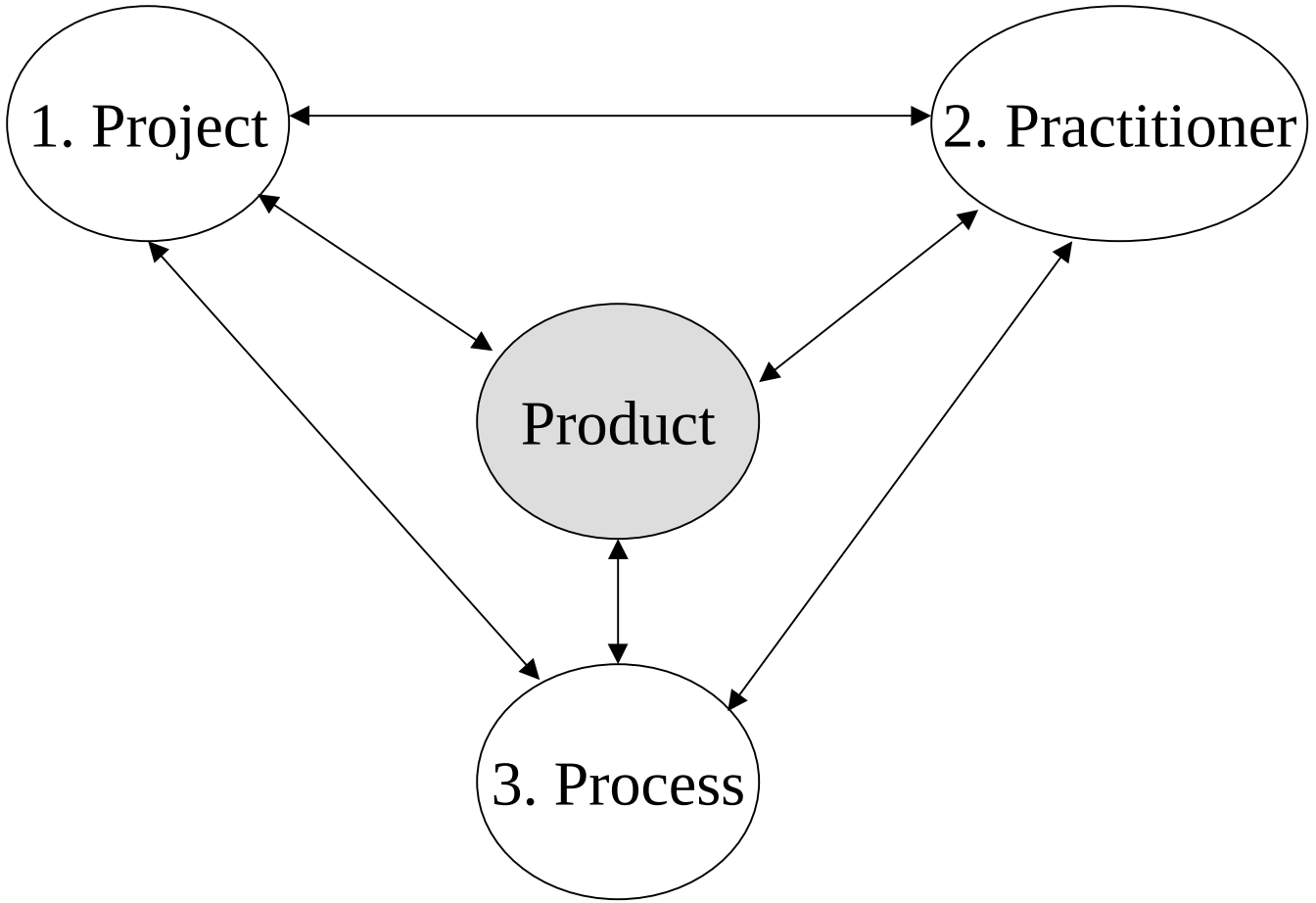
II. Draft - NASA Core Metrics

- NASA Core Software Metrics Report, Rev. 1, Jan 2000
- Contributing Organizations
  - Goddard SEL
  - Goddard SATC
  - NASA Software Working Group (Metrics Subgroup)
- Divided into 7 metrics groups
- Each metric group contains primary and secondary metrics

II. Draft - NASA Core Metrics

- Metric Groups
  - Defects
    - Severity, Where found, When found, When fixed, ...
  - Modifications
    - What type of mod, Why, When identified, When Fixed, ...
  - Requirements
    - Number of Requirements, Number verified vs. time, ...
  - Schedule
    - Planned vs. actual milestone dates, ...
  - Effort
    - Planned vs. actual effort (by subproduct, phase, …)
  - Code
    - Estimated code size, Actual code size, complexity measures, ...
  - Project (characteristics)
    - Project ID, Application type, Language, Life cycle model, platform, ...

II. Workshop 3a: Reminder - Business Goals are from three perspectives



II. Workshop 3a

- Exercise: Prioritize a subset of the NASA Core Metrics that match the Business Goals from Workshop 1
- Table/Scorecard is supplied *(handout)*
- Columns
  - 1. Mapping of Business Goals to Metrics (3 Perspectives)
  - 2. Prioritization of Metric (3 = Highest, 2=Med, 1=Low)
- Blank rows are provide on the last sheet to include any additional metrics
- Work in Groups or Individually
  - We would like to collect completed sheets to compile composite viewpoint on Metrics ranking

II. Key Metrics Resources used in MSP's consideration of a metrics set

- JPL Policy 'Project Resource Management' Rev.1 Dated Mar. 06, 1999
- (NASA Core) - NASA Core Software Metrics Report
- JPL D-15378 Rev. D, The JPL Software Development Process Description
- (CMM-L2) - SEI Technical report CMU/SEI-92-25, Sep. 1992
- JPL D-8431 Ver. 2.2, Project Measures, Apr. 1991
- The Practical Software Measurement Program (http://www.psmsc.com/)

II. MSP Metrics Selection Process

Selecting Indicators

- Generated a scorecard for the indicators to develop a selection criteria
  - Selected required indicators from the JPL policy
  - Selected indicators that were at least on three of the five documents
  - Made sure that there was at least one indicator defined and selected for each of the six software common issues even if there was no recommendation from the selected documents
- For the resulting selected indicators, described their objective and information needed to generate them.

II. MSP Metrics: Schedule and Progress

Milestone Performance

Objective: Identify the current status of major project events to allow assessment of the impact of potential or actual schedule slips on future activities and milestones.

Questions: Is the current schedule realistic? What is the projected completion date? Which milestones will be affected by the delay?

Indicators: Milestones planned and actual vs. time; list of delayed milestones for each reporting period

Collection Method: Extracted from a scheduling tool like MS Project, Timeline, Primavera, etc.

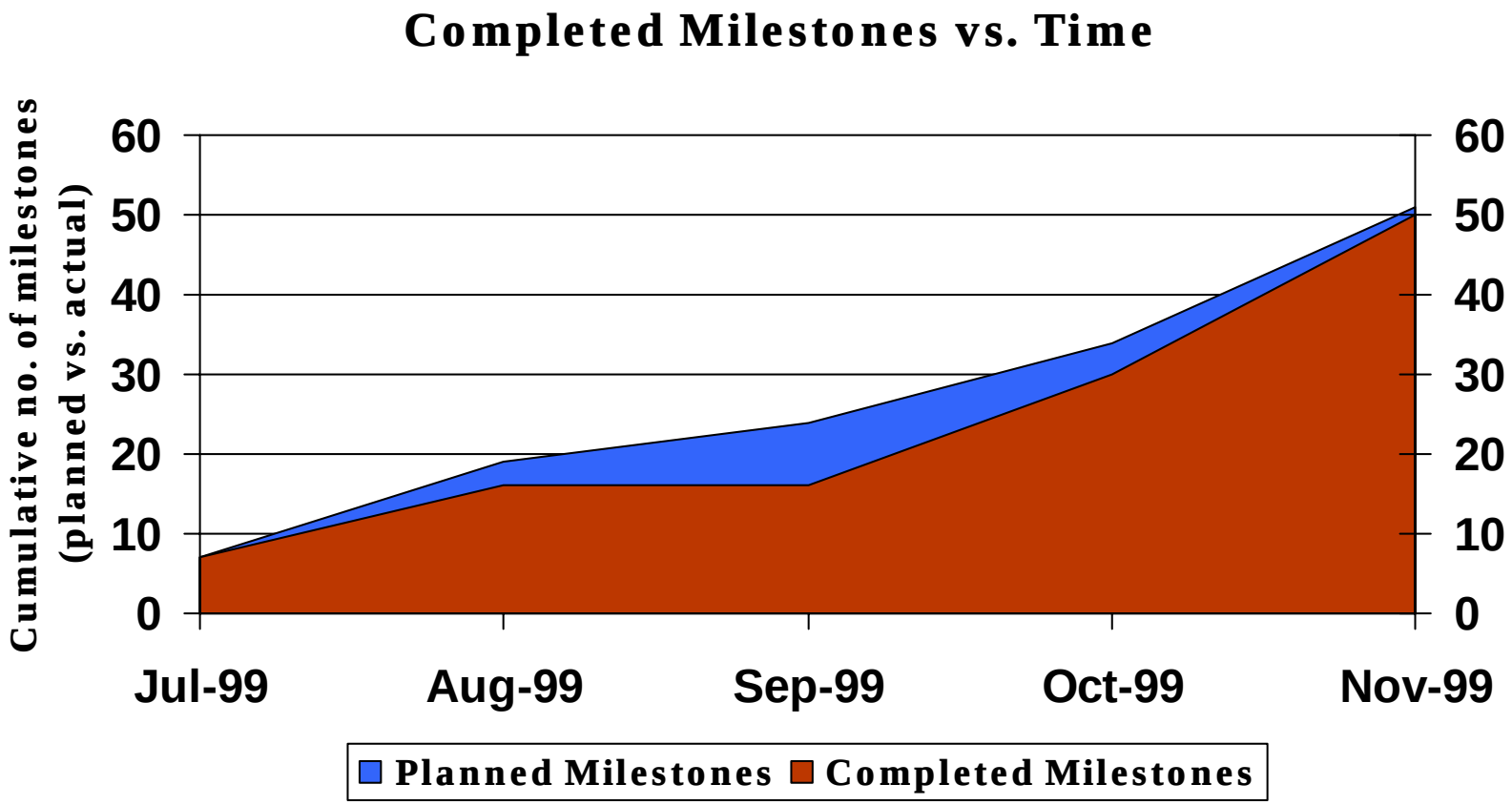
Supporting Activities: Project scheduling and periodic status schedule updates.

Other Option: Collect Work Element completion instead of milestones or both. Work elements are tasks from a WBS or segments of work as described by the project.

Recommended by: NASA Core, CMM-2, JPL Project Resource Management Policy , JPL D15378, PSM, and JPL D-8431

Applicability: Milestone Performance measures are applicable to all sizes and type of projects.

II. Schedule and Progress



II. MSP Metrics: What is Next?

- Get consensus on a measurement set.
- Provide a good definition of what is being measured and for what reason(s)
- Select tools to collect measures including standard indicator reports
- Provide basic indicator analysis
- Merge the collection and reporting of measures into the Tailored Process
- Include the measurement set in the pilots for Tailored Processes, Flight S/W Requirements, and Flight S/W Peer Review

III. Focused or Specialized Metrics

These are metrics sets which address specific processes or products, but do not attempt to address a wide spectrum of software concerns.

- Examples:
  - Software Inspection Metrics
  - Orthogonal Defect Classification
  - Reliability Estimation Metrics

III. Example: Software Inspection Metrics

Software Inspection is a well-defined technical peer review process for finding and fixing defects

- **Conducted by small team of peers with assigned roles**
- **Each participant has vested interest in work product**
- **Held within pre-test development phases on portions of engineering products**
- **Checklists are used to improve quality**

- **Metrics**
  - **Defects (number, type, severity)**
  - **Size of product**
  - **Time/Effort**
  - **Team Size and Composition**

http://eis.jpl.nasa.gov/quality/Formal_Methods/document/inspbrief.ppt

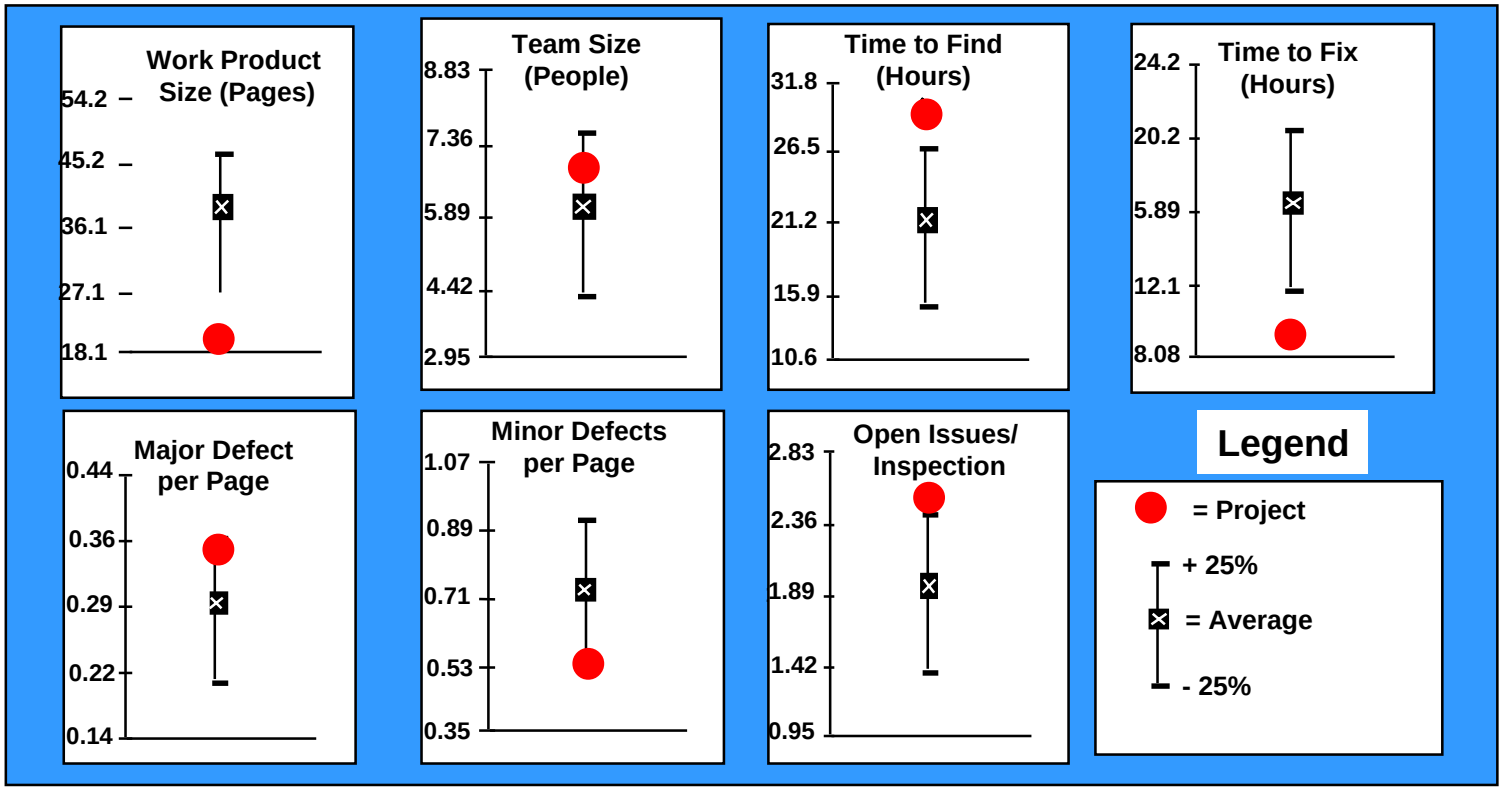
III. Example: Software Inspection Metrics

Inspection Metrics can be used to:

- Monitor Inspection Process
  - Meetings too Long
  - Preparation too Short
  - Size of Document too Large
- Look for Troubled Areas in Development
  - Large Number of Major Defects
  - Concentration of Specific Type or Category of Defects
  - Large Number of Open Issues
- Monitor Inspection Costs
- Assist in Tailoring Inspections
- Assist in planning where to use inspections, estimating cost, etc.

III. Example: Software Inspection Metrics

Project Report



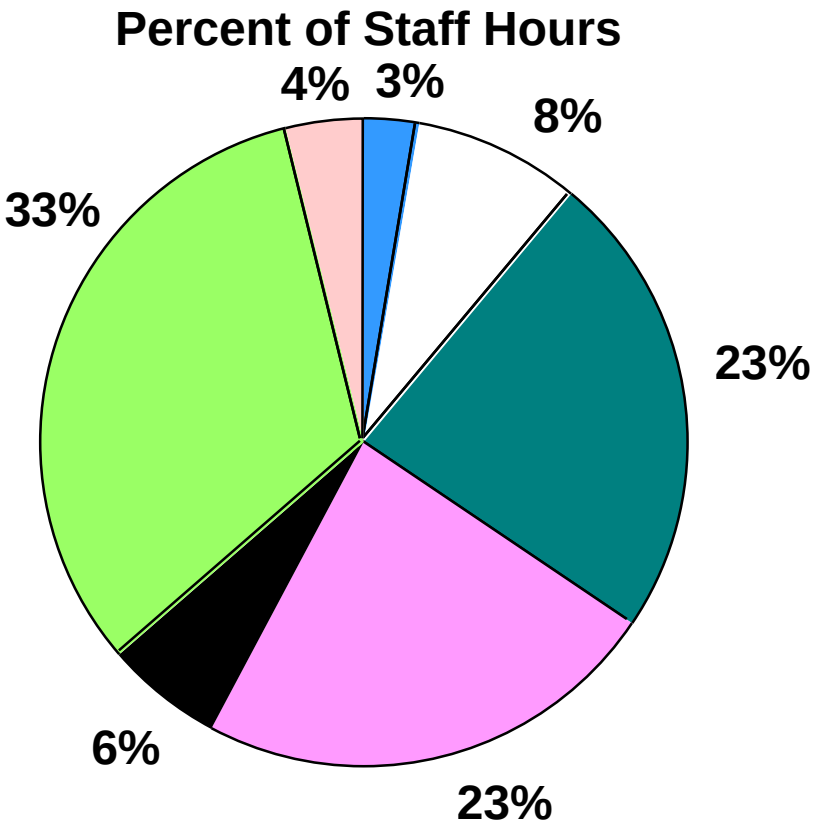
	Document (pages)	Team Size (people)	Time to Find (Hr)	Time to Fix (Hr)	Major Defects per Page	Minor Defect per Page	Open Issues per Inspection
JPL Averages	36.14	5.89	21.21	16.16	0.29	0.71	1.89
Project	20.45	6.64	31.18	8.68	0.37	0.60	3.18

Time to Find = Total Time Spent in Planning, Overview, Preparation and Meeting

Time to Fix = Total Time Spent in Rework, Third-Hour and Follow-Up

III. Example: Software Inspection Metrics

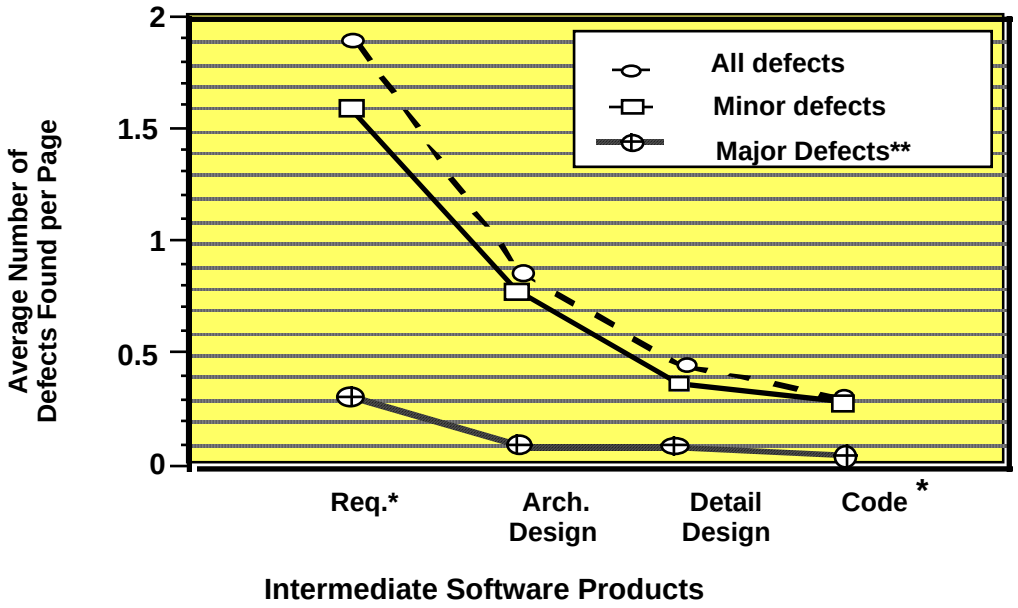
Effort: Hours per Stage of Inspection Process



Guidelines for Total Hours

Inspection Stages
Planning - 1 to 3 Hours
Overview - 1 to 5 Hours
Preparation - 2 Hours/Inspector
Meeting - 2 Hour Meeting
Third Hours - 0.5 to 3 Hours
Rework - 5 to 20 Hours
Follow-Up - 1 to 3 Hours

III. Defects Found by Software Inspections across Development Phase



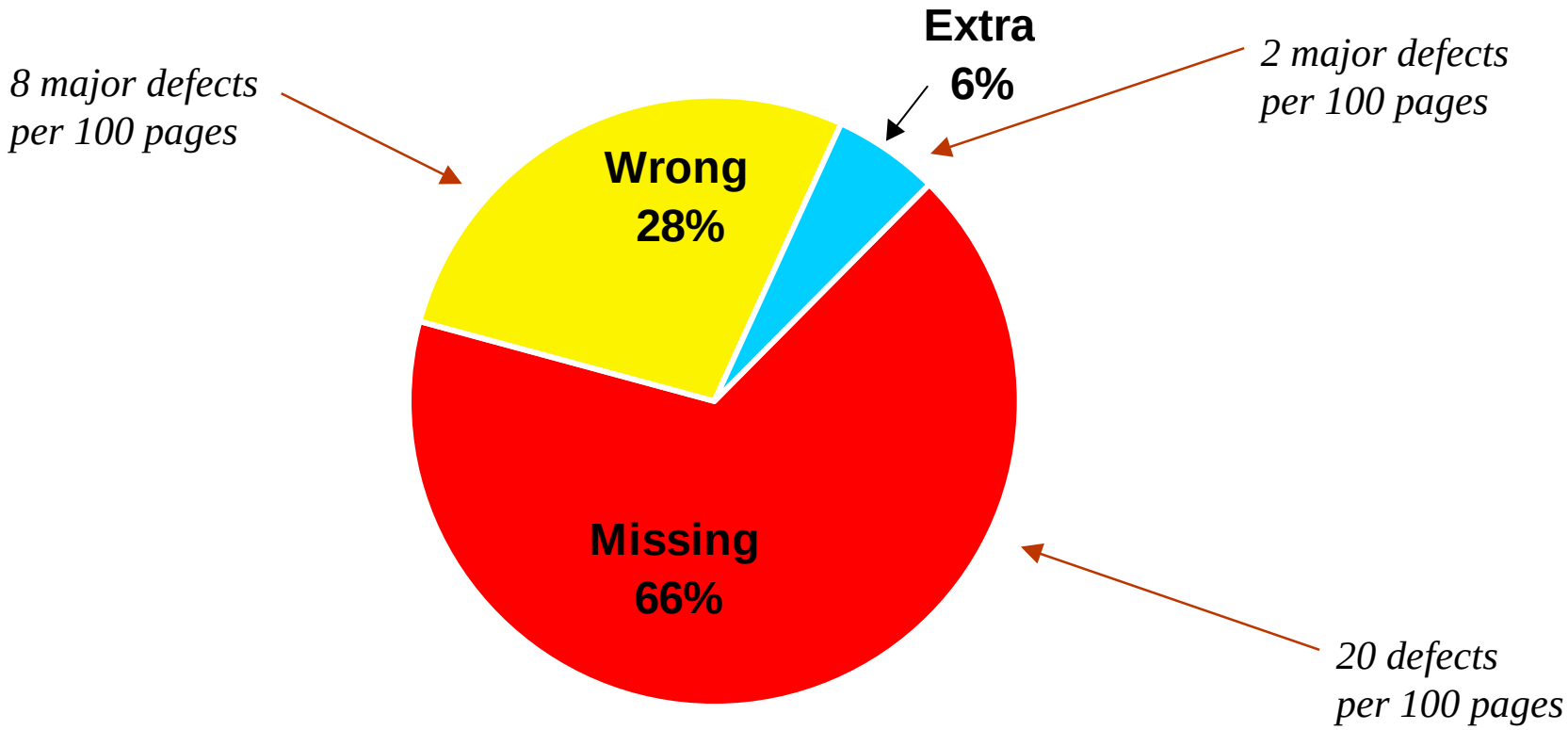
* At the alpha=0.05 level of significance ANOVA F test showed a *significant difference* between the defect densities found in Requirements and Code Inspections. This analysis is based upon a sample of 203 inspections performed on 6 JPL projects.

** **Definitions**: A **major defect** is an error that would cause the system to fail during operations, or prevent the system from fulfilling a requirement. **Minor defects** are all other defects which are non-trivial. **Trivial defects** in grammar and spelling were noted and corrected, but not included in this data analysis.

Requirements & design are high leverage phases to use Software Inspections

[Kelly92]

Distribution of Major Software Requirements Defects found during Software Inspections



Completeness is potentially the biggest quality issue for software requirements

Note: This is across a sample of 45 formal inspections of software requirements.

IV. Workshop 3b: Outlining a Metrics Strategy

- Objective: Outline a strategy for implementing software metrics programs with the assistance of Goddard & JPL "championing groups" (i.e. SEL, SATC, & MSP)

  1. Spectrum of implementation decisions (slide #27)

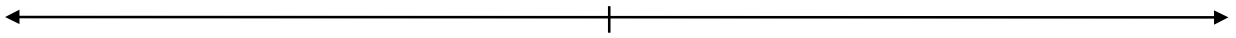
  2, Confronting the obstacles to implementing metrics (slide #28)

  3. A few key questions to explore which segments of NASA can aid in the implementation of software metrics (slides #29-31)

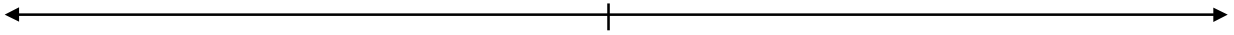
  4. Planning a sequence of actions/events for a successful implementation of software metrics (slide #32)

IV. Workshop 3b-1: Spectrum of implementation decisions

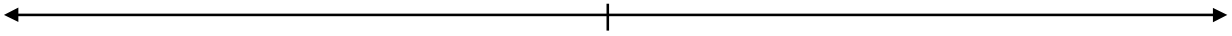
A) Top down vs. bottom up (__% / __%)



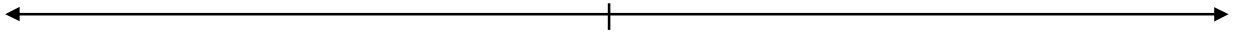
B) Evolution vs. Revolution (__% / __%)



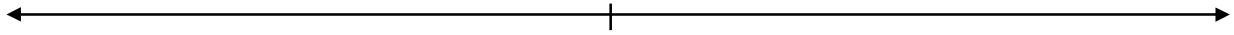
C) Local tailored metrics sets vs. NASA universal set (____%)



D) Access to metrics: Open vs. Controlled (__% / __%)



E) Reporting: Hierarchical vs. Flat (__% / __%)



Workshop 3b-2: Confronting the obstacles to implementing metrics

Obstacles:

a) What are the key obstacles to implementing metrics?

b) Why are these items obstacles?

c) What may remove or lower these obstacles?

Workshop 3b-3: A few key questions

a. What is the weakest link for successful used of metrics within our organizations? (Metrics Research, Tools, Discipline, Time, Culture, …)

b. Who are the key customers for software metrics? Are the “start-up” or “tech transfer” customers different from the operational customers?

Workshop 3b-3: A few key questions

c. Which organization should receive the primary responsibility and the funding for keeping, analyzing, and reporting metrics?

d. How do we get beyond the “Build it and they will come” paradigm?

Workshop 3b-3: A few key questions

e. Are we lacking the right set of metrics for software?

f. Do you anticipate a major “break-through” with respect to metrics in the next 5 -10 years? If so, what will it be?

Workshop 3b-4: Planning a sequence of actions/events for a successful implementation of software metrics

What are the sequence of events needed to make Software Metrics a practical reality at Goddard and JPL?

a.

b.

c.

d.

e.

…

Goddard Space Flight Center

JPL

Back-up Slides

MSP Metrics Selection Results
Scorecard

Indicators Organized By Common Issues	JPL Policy	NASA Core	JPL D-15378	EIS CMM-L2	JPL D-8431
Schedule and Progress					
1) Schedule milestone planned and actual vs. time	R	**X**	**X**	**X**	**X**
2) Requirement validation planned and actual vs. time	**X**	**X**			**X**
Resources and Cost					
3) Estimated and actual cost vs. time (total or by s/w activity?)	R		**X**	**X**	**X**
4) Total full time equivalent (FTE) work allocation (days/months) planned and actual vs. time	R	**X**	**X**		**X**
Growth and Stability					
5) Requirement volatility (based on requirements added, deleted, modified)			**X**	**X**	**X**
6) Software size, estimated and actual vs. time (could use function point, module count, code size, file size)		**X**	**X**	**X**	**X**
Product Quality					
7) Aging of defects/issues			**X**	**X**	**X**
8) Problem report trends (also used to determine product stability)		**X**	**X**	**X**	
Development Process Performance					
9) Defect/Problem report origin vs. finding activity			**X**		
Technical Adequacy					

Metrics Session

Independent Verification and Validation (IV&V) Facility

Ted Hammer

May 16 -18, 2000

*This presentation will be distributed at the workshop.